标题：基于Pandas+Seaborn+Matplotlib的居民健康状况数据可视化

摘要：

随着现代医疗水平和现代计算机科学、人工智能、数据分析的不断发展，医院里也积累了大量的病人病历信息数据。通过大量信息的整合和修改，对这些数据进行整理与分析，有助于优化医疗资源配置、提升服务质量以及预测患者治疗效果。本数据集汇总了多位患者的基础信息、病情描述、就诊过程及医疗费用等关键内容，涵盖字段包括姓名、年龄、性别、血型、入院与出院日期、主治医生、所在医院、保险公司、账单金额、住院时长等[1]。本研究旨在使用Python支持库Pandas+ Seaborn+ Matplotlib，通过数据分析，采集该数据集探索患者特征与医疗行为之间的潜在关系，分析例如年龄与账单金额的相关性、不同医疗条件下住院时长的差异等。通过数据分析及机器学习寻找不同字段之间的相关性，为智能医疗决策提供数据支持。结果表明：结果表明，通过对患者基础信息与医疗过程数据的深入分析，可以发现多种潜在的规律与关联关系。

Abstract：

With the advancement of modern medical technology, computer science, artificial intelligence, and data analytics, hospitals have accumulated a vast amount of patient medical record data. By integrating and refining this information, data analysis can help optimize the allocation of medical resources, improve service quality, and predict patient treatment outcomes. This dataset compiles key information from multiple patients, including basic demographics, disease descriptions, treatment processes, and medical expenses. It covers fields such as name, age, gender, blood type, admission and discharge dates, attending physician, hospital, insurance provider, bill amount, and length of stay. This study aims to explore the potential relationships between patient characteristics and medical behavior using Python libraries such as Pandas, Seaborn, and Matplotlib. Analyses include, for example, the correlation between age and billing amount, as well as differences in hospitalization duration under varying medical conditions. By applying data analysis and machine learning techniques, the study seeks to identify correlations among different variables and provide data-driven support for intelligent medical decision-making. The results show that in-depth analysis of patient demographics and medical process data reveals multiple underlying patterns and associations.

关键字：

0 引言

在计算机编程中，pandas是用于数据操纵和分析的Python软件库。它建造在NumPy基础上，并为操纵数值表格和时间序列，提供了数据结构和运算操作。[2]在Python中，pandas库的功能十分强大，它提供高性能的矩阵运算；可用于数据挖掘和数据分析，同时也提供数据清洗功能；支持类似SQL数据库的增、删、查、改等操作，并且带有丰富的数据处理函数；支持时间序列数据分析功能；支持灵活处理确实数据等。[3]Seaborn是构建在matplotlib之上的数据可视化库，与Python中的pandas数据结构紧密集成。可视化是Seaborn的核心部分，可以帮助探索和理解数据。[4] Matplotlib的功能则与Seaborn接近，也是帮助将数据进行可视化操作的第三方库函数。

随着科技的发展，我们每个人都是信息化时代的受益者。从医疗卫生的角度，医院积累了大量患者的诊断和就诊数据。我们对这些就诊数据进行数据分析和数据可视化，分析各个字段之间所存在的潜在联系。

综上所述，为了分析患者的客观信息及其医疗行为之间的关系，我们需要对此数据集进行数据分析，这里采用python中常见的第三方库，分别是Pandas, Seaborn, Matplotlib。

1 Pandas, Seaborn, Matplotlib 中各函数的功能、数学基础和基本原理

2 基于Pandas, Seaborn, Matplotlib 的数据分析

我们这次将从多个方面对该数据文件进行分析

首先我们会通过info观察csv文件的形状，存有的字段和数据类型。

然后，我们可以通过Matplotlib对文件进行预处理，一方面可以去尝试判断该文件中是否有字段出现缺失，另一方面，我们需要对可能存在的重复值进行去重操作。

随后，我们绘制相关系数热力图，试图从中寻找可能存在关系的字段，但是由于本数据集中的数值型数据较少，所以通过sns.heatmap函数无法有效绘制非数值型字段之间的热力图。

之后我们可以对在之前相关系数热力图分析中，对出现的可能存在关系的字段进行数据分析。数据分析可以从如下几个维度进行：绘制散点图、箱线图、柱状图等；使用Chi2进行独立性检验；建立模型以通过已有的字段预测其他字段。

最后我们利用scikit-learn中的支持向量机SVM，构建基于医疗数据集的支持向量机分类模型，并对该模型进行评价。

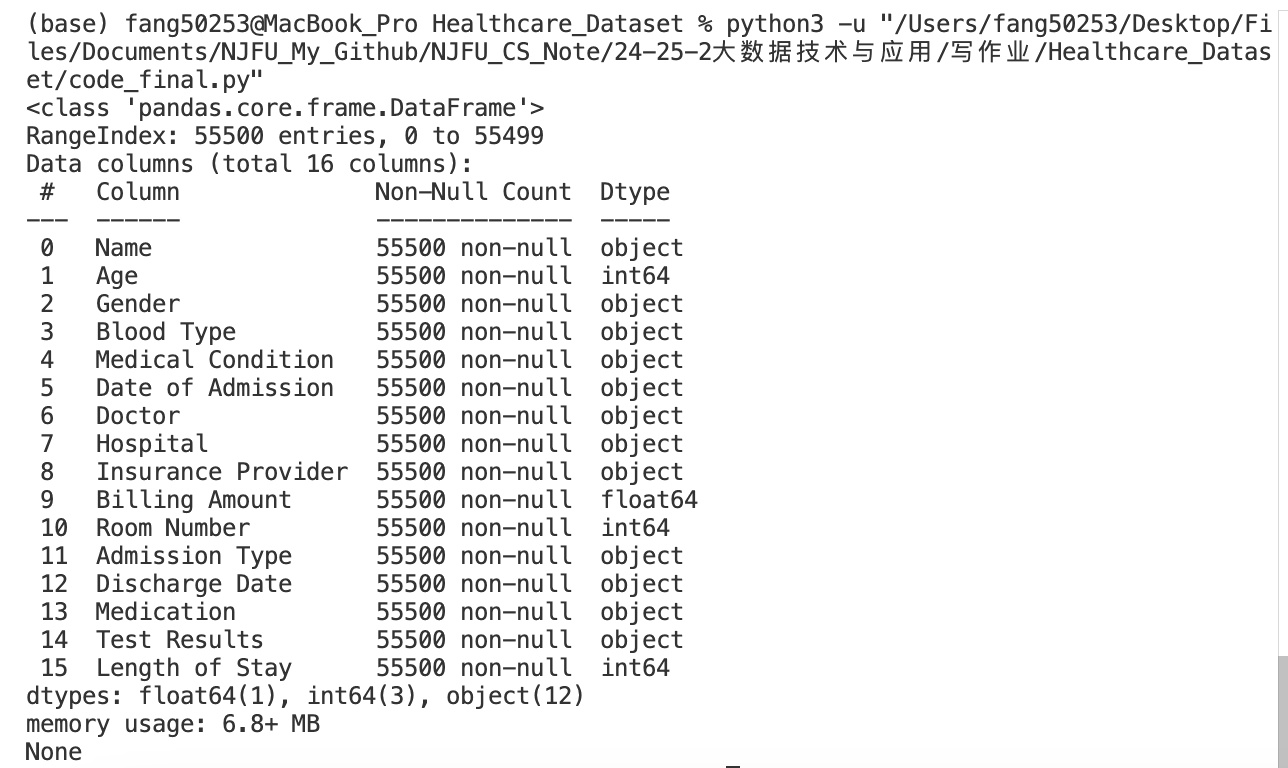
3 利用python对居民的健康数据进行数据预处理

收集到的数据存在如下字段。

| **英文字段（English）** | **中文字段（简体中文）** |
| --- | --- |
| Name | 姓名 |
| Age | 年龄 |
| Gender | 性别 |
| Blood Type | 血型 |
| Medical Condition | 病情情况 |
| Date of Admission | 入院日期 |
| Doctor | 主治医生 |
| Hospital | 医院 |
| Insurance Provider | 保险公司/保险提供方 |
| Billing Amount | 账单金额 |
| Room Number | 病房号 |
| Admission Type | 入院类型（如急诊/预约等） |
| Discharge Date | 出院日期 |
| Medication | 药物/用药情况 |
| Test Results | 检查结果 |
| Length of Stay | 住院时长 |

* 数据导入

首先导入数据集并使用info查看具体数据，程序运行结果如图所示：



首先大致观察数据和列标签的形态，接着运用相关函数查看数据的基本信息。由此可知，该数据集共包含55,500行、16列，记录了医院接诊的大量患者的患病和治疗信息其中，“Billing Amount”一列为浮点型，表示患者的账单金额；“Age”、“Room Number”以及“Length of Stay”为整型，分别记录了患者的年龄、住院病房号和住院的时长；其余 12 列均为对象（object）类型，对应字符串数据。例如，“Date of Admission”和“Discharge Date”表示患者的入院和出院日期；“Gender”、“Blood Type”、“MedicalCondition”等列提供了患者的基本信息；“Doctor”和“Hospital”表示接诊医生及其所在医院；而“Insurance Provider”、“Admission Type”、“Medication”和“Test Results”等则涵盖了患者的医保类型、入院方式、用药情况及检测结果等。例如，数据集中采用保险公司的具体名称，比如UnitedHealthcare、Blue Cross、Aetna、Cigna、Medicare来表示保险的提供商。

最后输出的该csv文件共占用存储空间6.8MB。

* 查看数据缺失情况

下面我们采用matplotlib.pyplot库中的绘图函数，绘制一张缺失值热图。显然，在这段数据中没有缺失值，如果存在缺失值我们还需要对缺失值进行补全操作。代码和绘制的图片如下：

# 2.使用Matplotlib查看是否存在数据缺失

plt.figure(figsize=(12, 6))

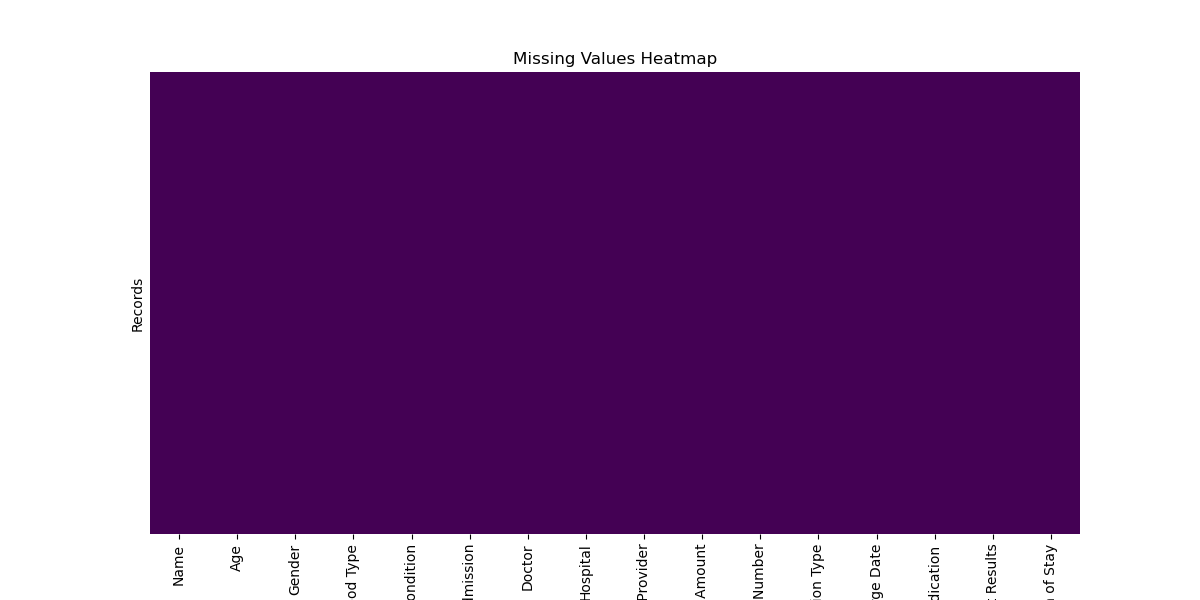
sns.heatmap(data.isnull(), cbar=False, cmap='viridis', yticklabels=False)

plt.title('Missing Values Heatmap')

plt.xlabel('Columns')

plt.ylabel('Records')

plt.savefig('missing\_values\_heatmap.png')



从最后输出的热力图上来看，所有字段的数据均未发生数据缺失，因此本数据的每一条记录都是相对完整的，无需对缺失值进行处理，或者插值。

* 重复值分析

重复值分析：经过分析后我们发现原数据不存在重复值，代码如下：

# 3.重复值分析

duplicate\_rows = data.duplicated()

num\_duplicates = duplicate\_rows.sum()

print(f"Number of duplicate rows: {num\_duplicates}")

if num\_duplicates > 0:

print("Duplicate rows:")

print(data[duplicate\_rows].head())

data\_cleaned = data.drop\_duplicates()

print(data\_cleaned.info())[5]

清除重复数据前，数据集的尺寸为55500行×16列；清除后，数据集的大小仍为55500行×16列，这表明本数据集没有重复的数据值，无需对重复值进行去重操作。

4. 利用python对居民的健康数据进行数据分析

* 绘制相关系数热力图

为了解各个字段之间的相关性，我们可以绘制相关系数热力图矩阵，通过观察各个字段之间的相关系数对数据做出初步分析。由于object类型的数据无法直接计算相关系数，因此我们使用numeric\_df = data\_cleaned.select\_dtypes(include='number')提取数值型的字段，对数值型的数字进行数据分析，代码如下：

# 4.数据相关系数热力图

plt.figure(figsize=(12, 6))

numeric\_df = data\_cleaned.select\_dtypes(include='number')

correlation\_matrix = numeric\_df.corr()

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f",square=True)

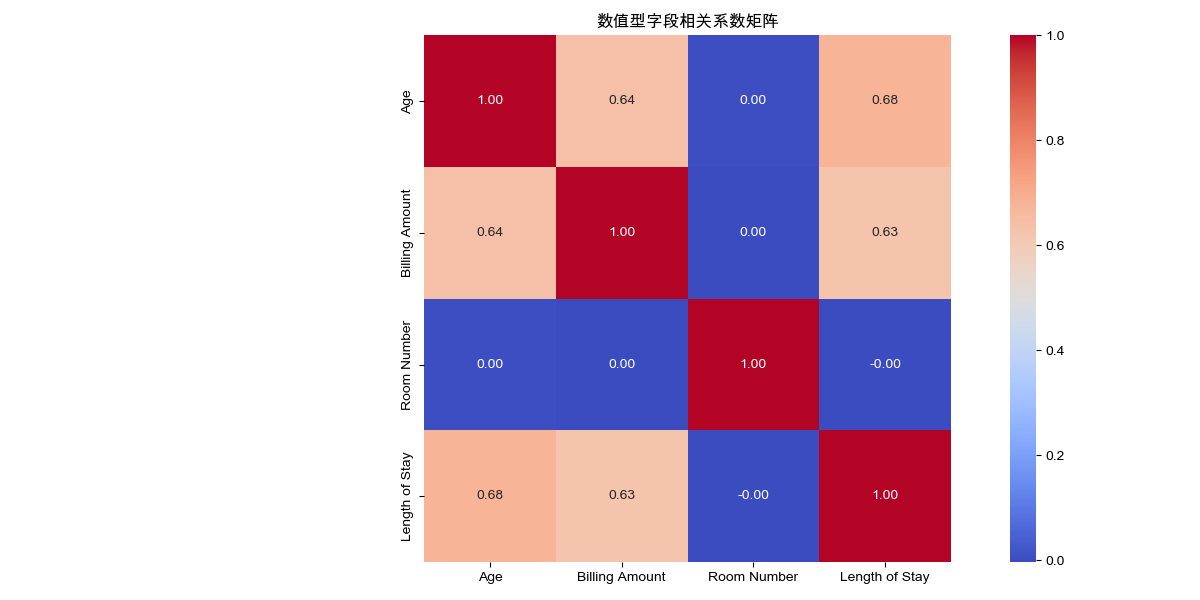
plt.title("数值型字段相关系数矩阵")

plt.tight\_layout()

plt.savefig('correlation\_matrix\_heatmap.png')

从图中我们看到Room，即房间号与年龄、住院时长和账单天数之间无相关性，符合常规的认知。年龄和账单金额、账单金额和住院天数、住院天数和年龄之间都有中等偏强的相关性，因此我们在下面的数据分析中可以着重分析这些具有较强相关性的字段之间的规律。

但是由于原始数据中的数值型数据数量较为有限，为了更加准确地展现各个字段之间的相关性，我们还需要想办法对非数值型的数据进行比较，以此来确定他们之间存在的联系。



* 统计账单金额箱线图

通过观察csv文件发现，这个数据集中的数据主要来源于4家医院，分别对这4家医院的账单金额进行统计，并绘制箱线图，代码如下：

# 5.将数据按照医院划分，统计账单金额的箱线图

plt.figure(figsize=(12, 6))

sns.boxplot(data=data\_cleaned, x='Hospital', y='Billing Amount')

plt.title('各医院账单金额分布（箱线图）')

plt.xlabel('医院名称')

plt.ylabel('账单金额')

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.savefig('billing\_amount\_boxplot\_by\_hospital.png')

quantiles = data\_cleaned.groupby('Hospital')['Billing Amount'].quantile([0.25, 0.5, 0.75])

quantiles = quantiles.unstack()

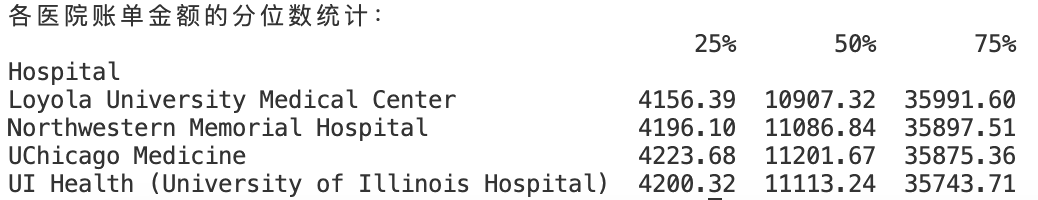
quantiles.columns = ['25%', '50%', '75%']

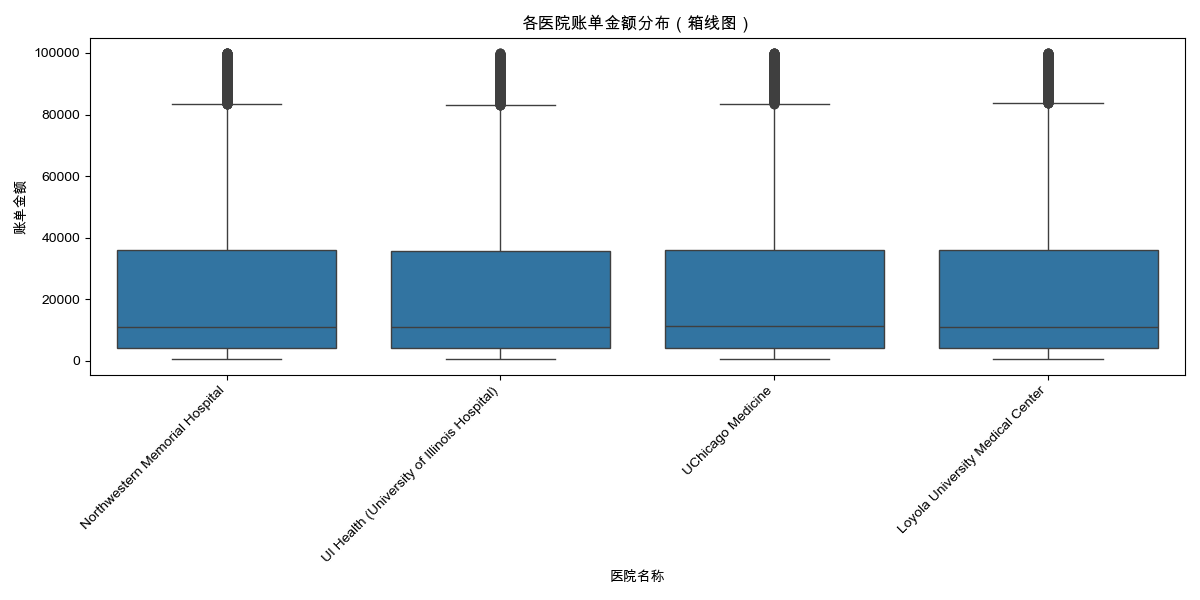
quantiles = quantiles.round(2)

print("各医院账单金额的分位数统计：")

print(quantiles)

我们得到了如下结果：



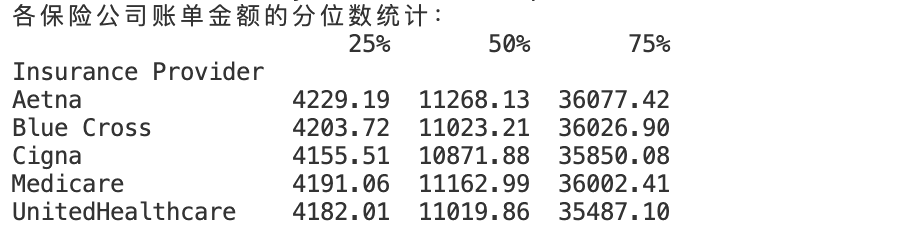


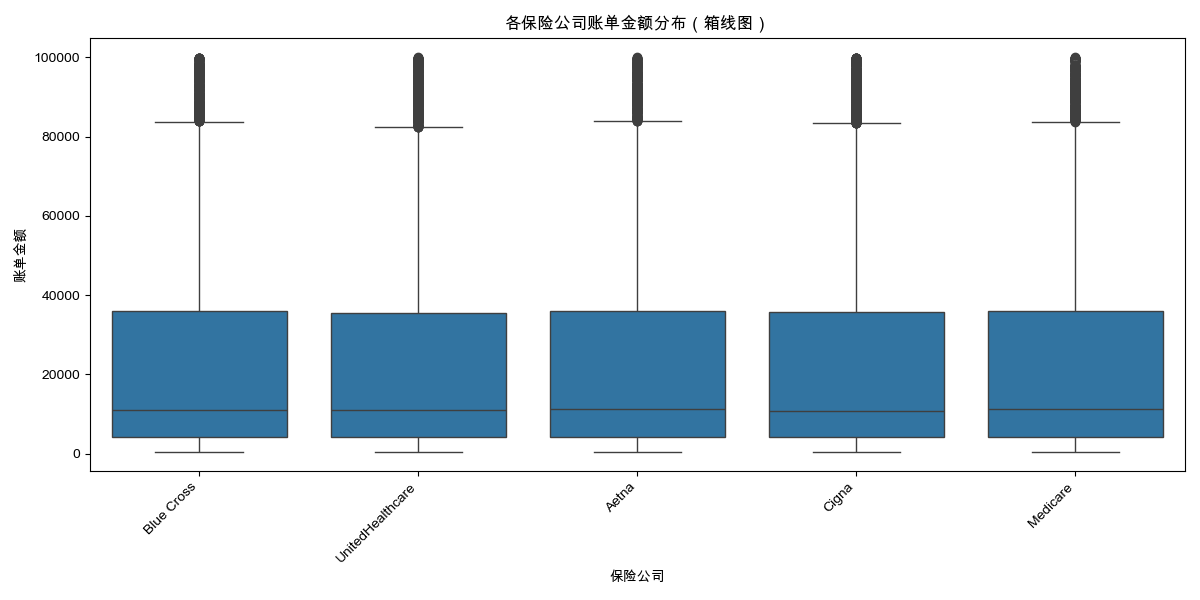
由箱线图可知，接受统计的各个4家医院在账单金额上接近，且账单金额的25、50、75百分位数也比较接近。没有表现出显著差异，而仅仅通过数值型数据的相关系数矩阵是很难体现出来的。

* 对不同保险公司的账单金额作箱线图

同样，我们可以对保险公司、检查结果和用药情况做出类似分析：

对于不同的保险公司提供商，其账单金额也没有表现出显著差异。

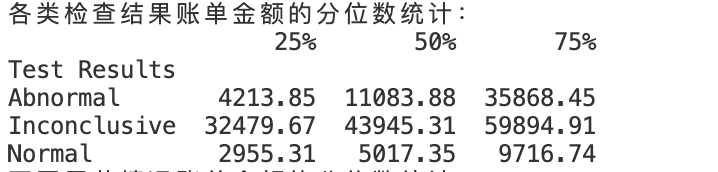


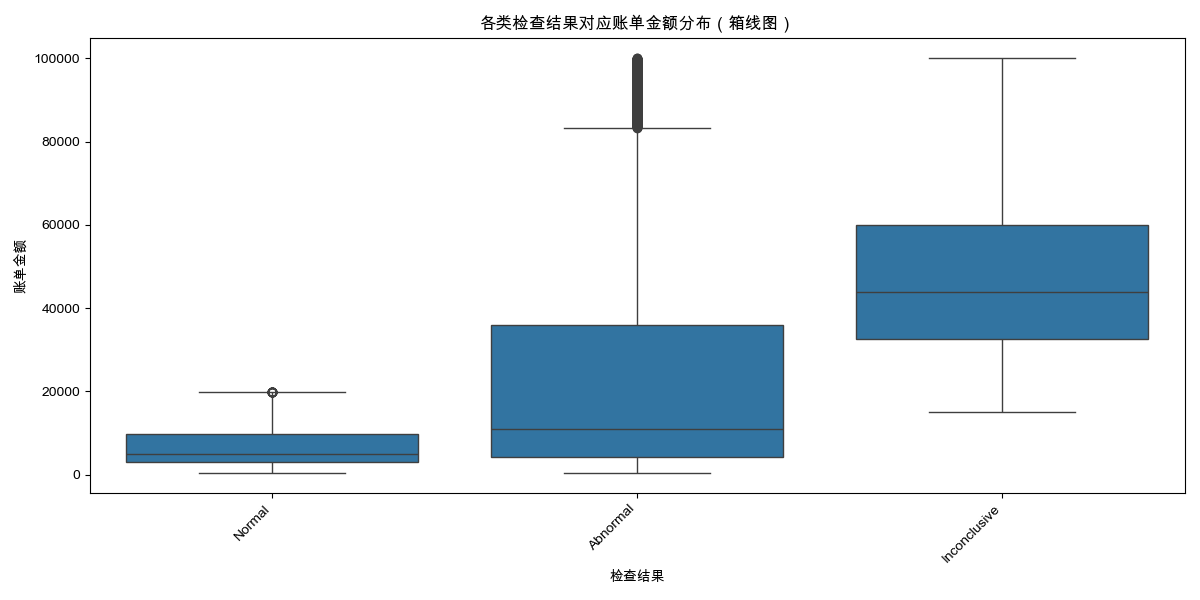


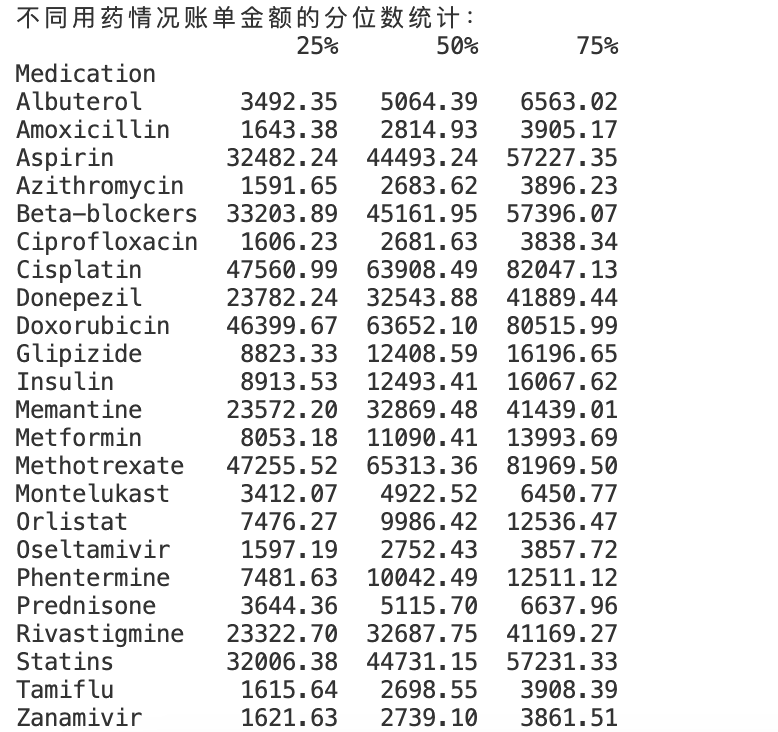
但是面对不同的检查结果，一般的检查结果和不正常的检查结果有着显著的差距，面对无法确诊的病症，其账单即费用远高于一般的病症，这也意味着患者或者保险公司将会花费更多的钱。

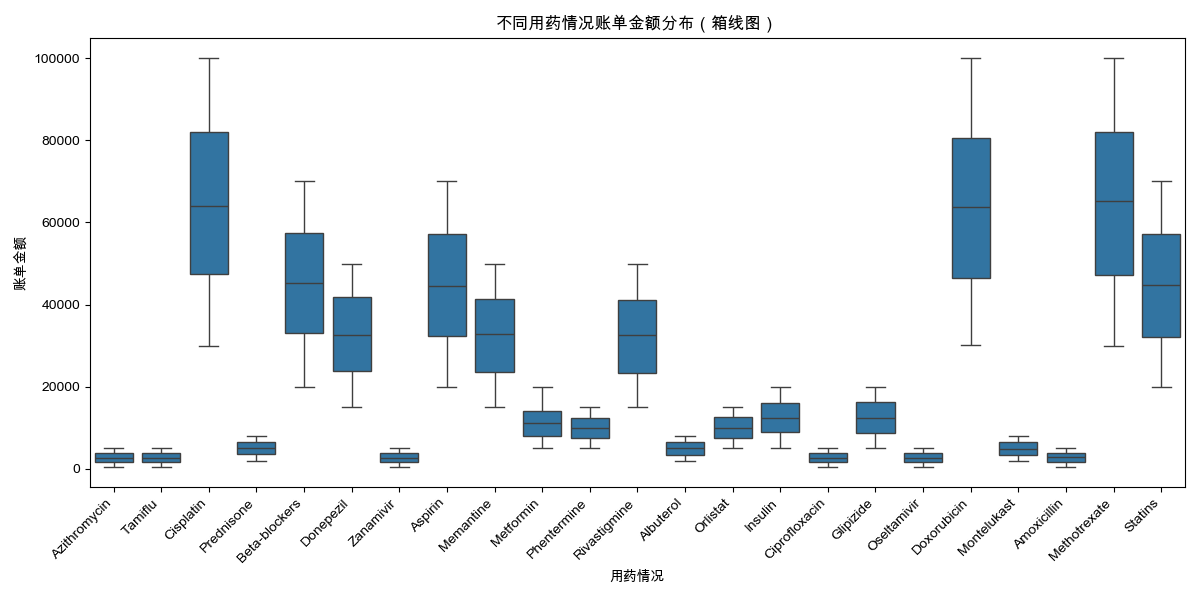
* 不同的用药情况和诊断难度对账单金额的影响

不同的用药情况同样会带来账单金额的巨大差异，部分药品的价格高昂，部分药品的价格低廉，从箱线图中我们也看到了不同药品之间的显著差异。









* 性别和患病病情之间的关系

下面我们来分析另2个可能存在关系的，即性别和病情之间的关系。这次我们可以选择对两者进行独立性检验，即利用卡方检验计算二者的相关性，并计算p值。但是最终我们发现p值高达0.8113，二者之间不存在显著关联。下面给出进行独立性检验的代码：

# 7.独立性检验性别VS病情

contingency\_table = pd.crosstab(data['Gender'], data['Medical Condition'])

print("\n交叉表（性别 vs. 病情）:")

print(contingency\_table)

chi2, p, dof, expected = chi2\_contingency(contingency\_table)

print("\n卡方检验结果:")

print(f"Chi2统计量: {chi2:.4f}")

print(f"自由度: {dof}")

print(f"p值: {p:.4f}")

alpha = 0.05

if p < alpha:

print("结论：性别与患病种类之间存在显著统计关联。")

else:

print("结论：性别与患病种类之间不存在显著统计关联。")

plt.figure(figsize=(10, 6))

sns.heatmap(contingency\_table, annot=True, cmap="YlGnBu", fmt="d")

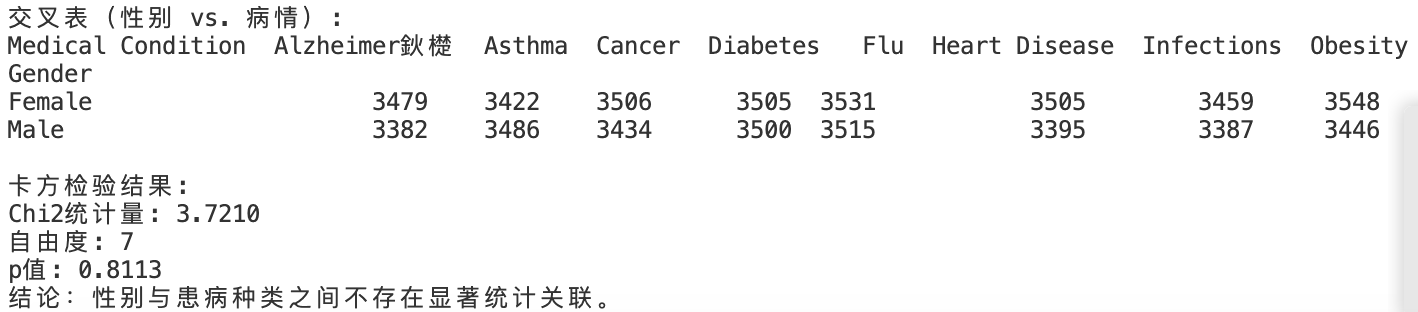
plt.title("性别 vs. 患病种类 - 交叉分布热力图")

plt.ylabel("性别")

plt.xlabel("患病种类")

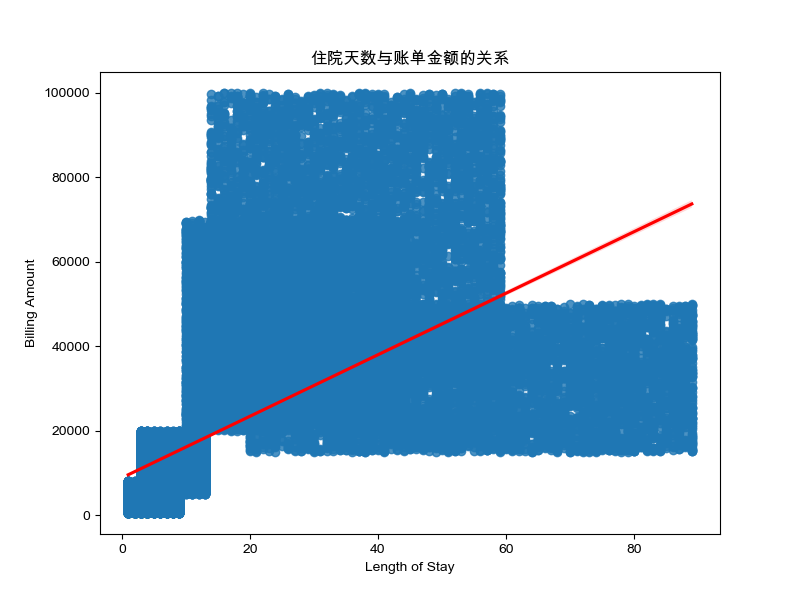
plt.tight\_layout()

plt.savefig("患病情况与性别的关系.png")



* 账单金额和住院天数之间的关系

接下来我们分析账单金额与住院天数的关系，不同于上面，这次我们采用散点图的方式来绘制，并且通过计算其回归方程。



* 不同疾病患病人数柱状图

我们可以分析不同疾病患病人数之间的差异。对于这种问题我们最好的选择是绘制柱状图，他是一种以长方形的长度为变量的统计图表。长条图用来比较两个或以上的价值（不同时间或者不同条件），只有一个变量，通常利用于较小的数据集分析。[6]而患病人数是这张图表中的唯一变量，且之间不存在随时间变化的关系，即在一段时间内的患病人数，因此适用于使用柱状图。分析所使用的代码如下：

# 9.不同疾病患病人数的柱状图（按性别分组）

disease\_gender\_counts = data.groupby(['Medical Condition', 'Gender']).size().unstack().fillna(0)

disease\_gender\_counts['Total'] = disease\_gender\_counts.sum(axis=1)

disease\_gender\_counts = disease\_gender\_counts.sort\_values(by='Total', ascending=False).drop(columns='Total')

plt.figure(figsize=(14, 6))

bar\_width = 0.4

index = np.arange(len(disease\_gender\_counts))

plt.bar(index - bar\_width/2, disease\_gender\_counts['Male'], width=bar\_width, label='男')

plt.bar(index + bar\_width/2, disease\_gender\_counts['Female'], width=bar\_width, label='女')

plt.xlabel('疾病名称')

plt.ylabel('患病人数')

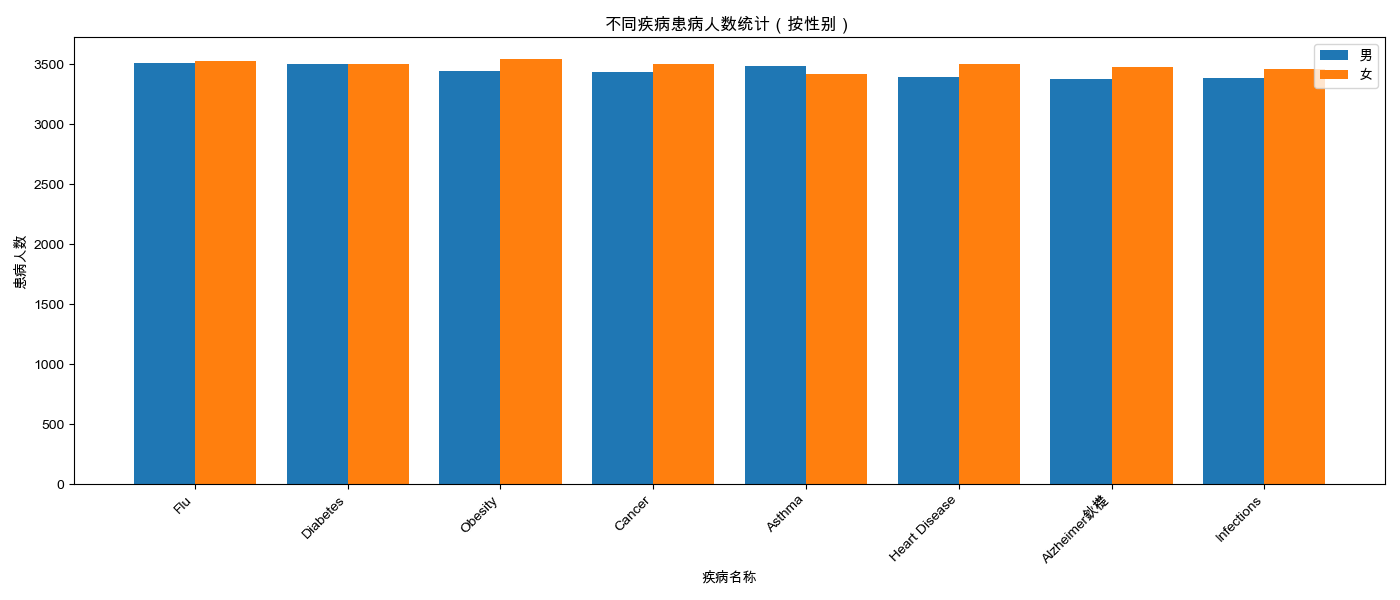
plt.title('不同疾病患病人数统计（按性别）')

plt.xticks(index, disease\_gender\_counts.index, rotation=45, ha='right')

plt.legend()

plt.tight\_layout()

plt.savefig("不同疾病患病人数统计\_柱状图\_按性别.png")



通过图表可以观察到，各疾病之间不存在显著的性别差异，患病人数也不存在显著的差异。

* 患病人数与实践之间的关系

data['Date of Admission'] = pd.to\_datetime(data['Date of Admission'], errors='coerce')

data\_valid\_dates = data.dropna(subset=['Date of Admission'])

data\_valid\_dates['Month'] = data\_valid\_dates['Date of Admission'].dt.to\_period('M').astype(str)

monthly\_counts = data\_valid\_dates['Month'].value\_counts().sort\_index()

plt.figure(figsize=(12, 6))

sns.lineplot(x=monthly\_counts.index, y=monthly\_counts.values, marker='o')

plt.title("每月入院人数统计（折线图）")

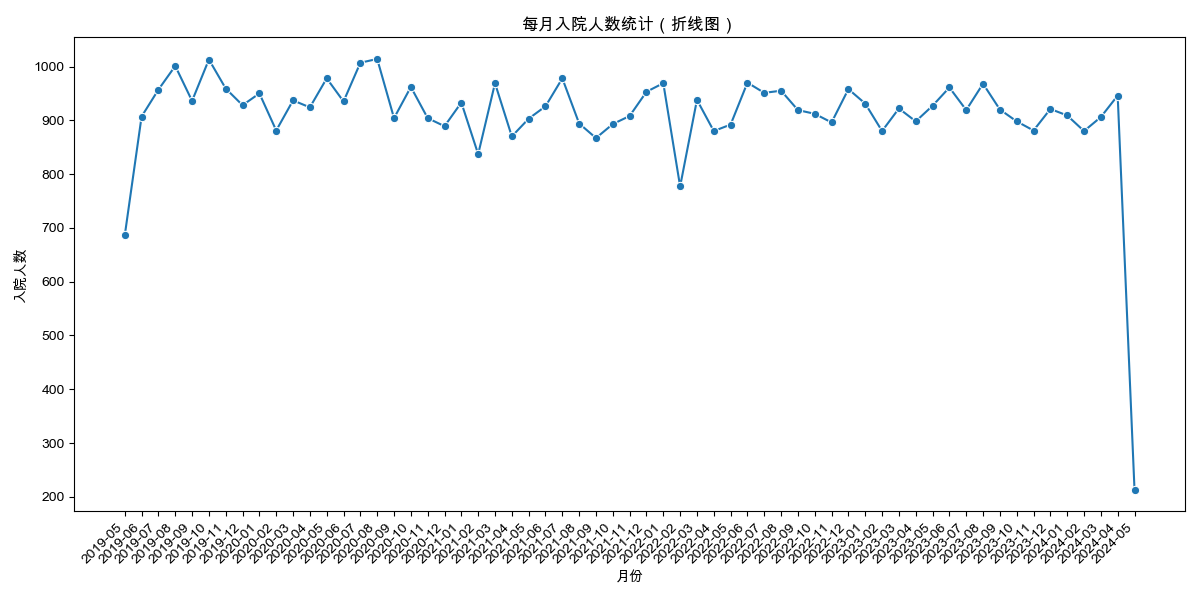
plt.xlabel("月份")

plt.ylabel("入院人数")

plt.xticks(rotation=45, ha='right')

plt.tight\_layout()

plt.savefig("每月入院人数折线图.png")



通过这张图片我们发现入院人数似乎随着时间的推移而逐渐减少，因此我们修改代码，在这张图上继续画出入院人数和时间的回归方程。

# 10.每月入院人数折线图（按Date of Admission统计）

data['Date of Admission'] = pd.to\_datetime(data['Date of Admission'], errors='coerce')

data\_valid\_dates = data.dropna(subset=['Date of Admission'])

data\_valid\_dates['Month'] = data\_valid\_dates['Date of Admission'].dt.to\_period('M').astype(str)

monthly\_counts = data\_valid\_dates['Month'].value\_counts().sort\_index()

monthly\_df = pd.DataFrame({

'Month': monthly\_counts.index,

'Count': monthly\_counts.values

})

monthly\_df['Month\_Num'] = np.arange(len(monthly\_df))

plt.figure(figsize=(12, 6))

sns.lineplot(x='Month\_Num', y='Count', data=monthly\_df, marker='o', label='每月入院人数')

sns.regplot(x='Month\_Num', y='Count', data=monthly\_df, scatter=False, color='red', label='趋势回归线')

plt.xticks(ticks=monthly\_df['Month\_Num'], labels=monthly\_df['Month'], rotation=45, ha='right')

plt.xlabel("月份")

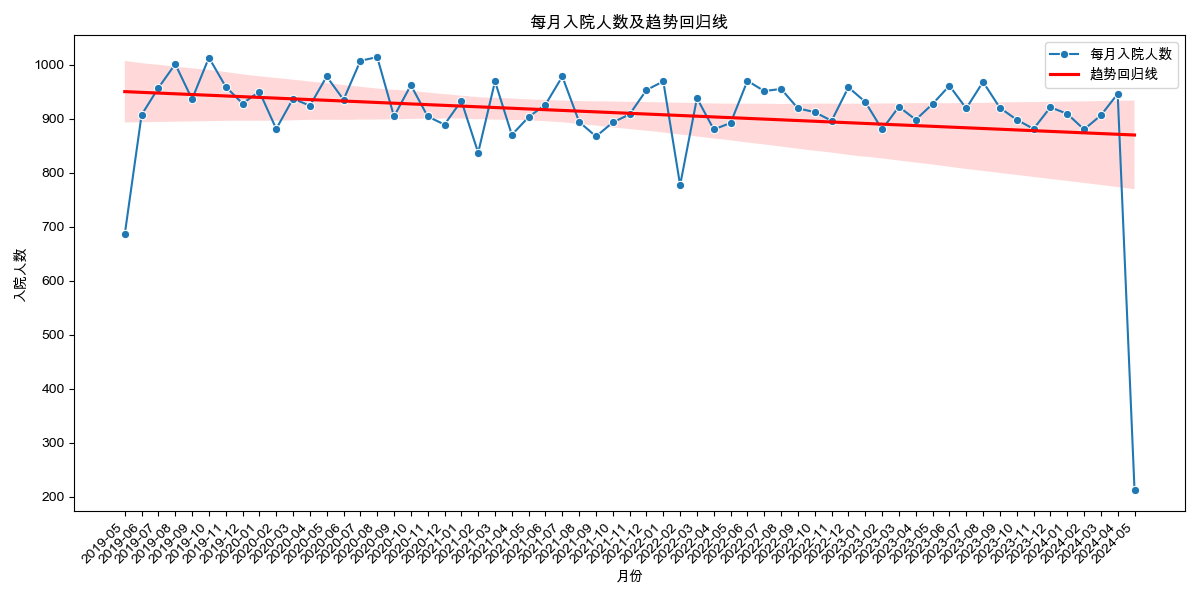
plt.ylabel("入院人数")

plt.title("每月入院人数及趋势回归线")

plt.legend()

plt.tight\_layout()

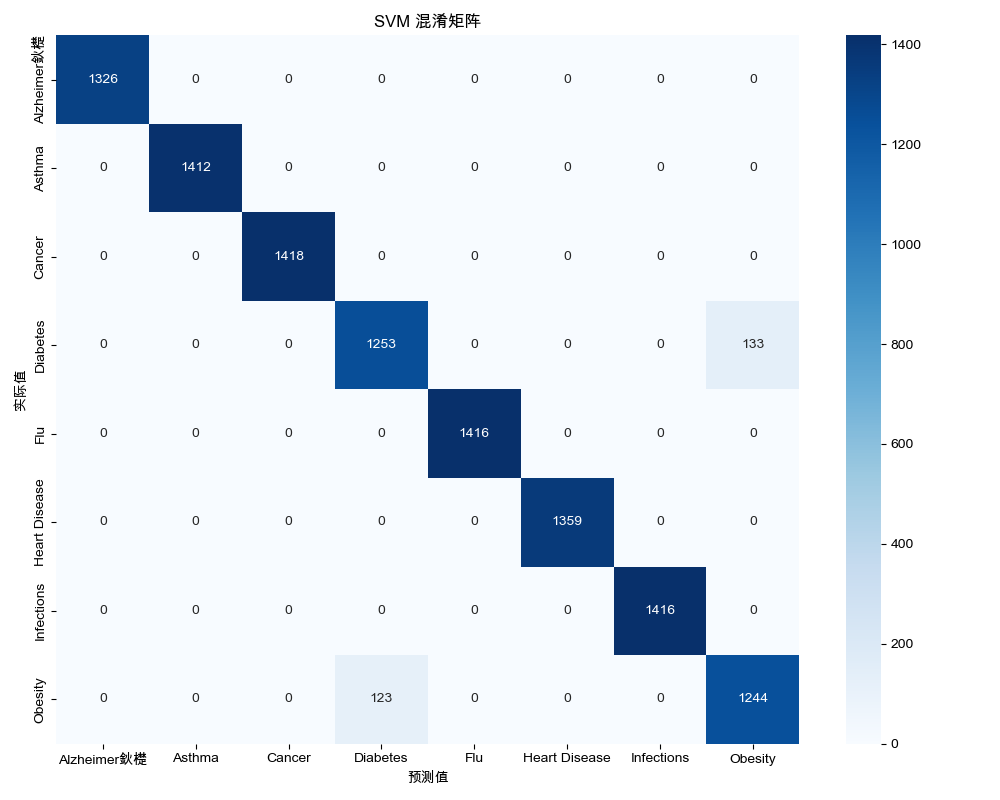
plt.savefig("10每月入院人数\_含趋势线.png")



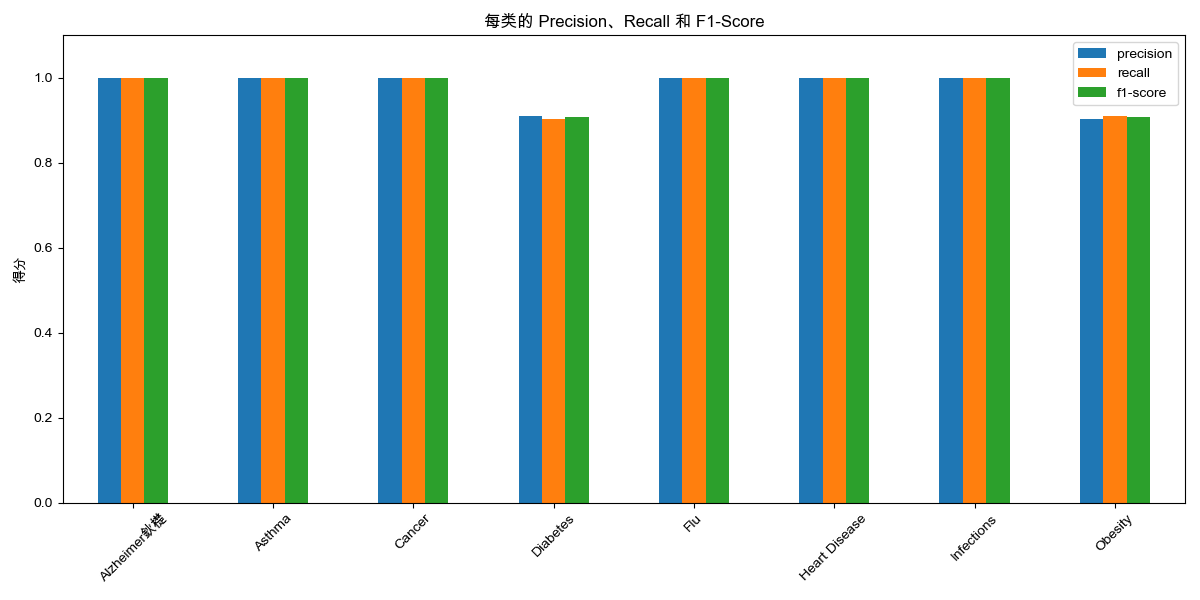
绘制的结果证实了猜想。随着时间的推移每月的入院人数有下降趋势，这可能是由多方面因素共同作用的结果。可能是地区人口减少、医疗水平的提高、居民健康情况改善、地区环境治理成效、医疗保险政策调整等多方面的。这当然是一个好的趋势，但如果我们需要分析具体原因，需要更多数据集的支持。

经过了上述10段数据分析，我们对医院在这段时间内对数据情况有了大致的了解。接下来我们想使用scikit-learn构建相应的模型，根据其他的数据来预测患各种疾病人数，帮助医院合理安排医疗资源。

* 对数据集进行划分，20%用于测试集，80%为训练集，使用支持向量机模型SVM构建分类模型。



* SVM支持向量机的评价



5 实验结果与分析

6 结论

参考文献

1. Eduardo Licea. Healthcare Dataset[EB/OL]. [2024-04-20]. <https://www.kaggle.com/datasets/eduardolicea/healthcare-dataset/data>.
2. 维基百科编者. Pandas[G/OL]. 维基百科, 2025(20250317)[2025-04-20]. <https://zh.wikipedia.org/w/inex.php?title=Pandas&oldid=86470885>
3. 曾文权, 张良均主编, 黄红梅, 施兴, 黄添喜副主编. Python数据分析与应用[M]. 第2版. 北京：中国工信出版集团，人民邮电出版社，2021：54.
4. deephub. CSDN2020(20200818)[2025-04-20]. <https://blog.csdn.net/deephub/article/details/108068984#:~:text=Seaborn%20%E6%98%AF%E4%B8%80%E4%B8%AA%E5%9F%BA%E4%BA%8Ematplotlib,%E8%81%9A%E5%90%88%E4%BB%A5%E7%94%9F%E6%88%90%E4%BF%A1%E6%81%AF%E5%9B%BE%E3%80%82>
5. [1]徐豪,刘婉月,张自豪.基于Pandas+Seaborn+Matplotlib的城市共享单车租赁分析可视化[J].现代信息科技,2024,8(23):58-62+68.DOI:10.19850/j.cnki.2096-4706.2024.23.013.
6. 维基百科编者. 条形图[G/OL]. 维基百科, 2024(20240912)[2024-09-12]. <https://zh.wikipedia.org/w/index.php?title=%E6%9D%A1%E5%BD%A2%E5%9B%BE&oldid=84180253>.